



# UNITED STATES PATENT AND TRADEMARK OFFICE

*SD*  
UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/790,668	03/01/2004	Richard E. Breault	TRCX-006/00US (300171-202)	7875
23419	7590	06/04/2007	EXAMINER	
COOLEY GODWARD KRONISH LLP			STEELEMAN, MARY J	
ATTN: Patent Group			ART UNIT	PAPER NUMBER
Suite 500			2191	
1200 - 19th Street, NW			MAIL DATE	
Washington, DC 20036-2402			06/04/2007	
			DELIVERY MODE	
			PAPER	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	10/790,668	BREault, RICHARD E.
	Examiner MARY STEELMAN	Art Unit 2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 01 March 2004.  
 2a) This action is FINAL.                            2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-48 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1-48 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on 01 March 2004 is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) Notice of References Cited (PTO-892)  
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3) Information Disclosure Statement(s) (PTO/SB/08)  
 Paper No(s)/Mail Date 10/12/2004.

4) Interview Summary (PTO-413)  
 Paper No(s)/Mail Date. \_\_\_\_\_.  
 5) Notice of Informal Patent Application  
 6) Other: \_\_\_\_\_.

## DETAILED ACTION

1. Claims 1-48 are pending.

### *Claim Rejections - 35 USC § 112*

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1, 14, 15, 30, 31, 41 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term "substantially" is a relative term which renders the claim indefinite. The term "substantially" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

### *Claim Objections*

3. Claim 23 is confusing. Claim may be amended to recite:

The system of claim 15, wherein the processor is a first processor, the conversion code includes encoded instructions to generate the first load module using the first processor, and the conversion code includes encoded instructions to generate the second load module using a second processor.

### *Claim Rejections - 35 USC § 102*

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an

international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-5, 10, 11, 13-19, 22, 23, 27-31, 39-42, 47 and 48 are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent Application Publication 2006 / 0101433 A1 to Opem et al.

Per claim 1:

A method for building a program for execution by a programmable controller, the method comprising:

- receiving a source program, the source program including high-level instructions for controlling a programmable controller;
- converting the source program to a first processor-executable program and a second processor-executable program; comparing the first and second processor-executable programs;
- sending, in response to the first and second processor-executable programs being substantially the same, one of the first and second processor-executable programs to the programmable controller.

Opem: [0013-0014], test program defined in a control language is compiled...compiler is validated, compiler is revalidated for errors introduced between the first and second compilation by comparing the first and second software means...functionality adds safety features to an industrial control system [0035], comparing software means and enabling the user written program...software means is downloaded to the device 6a (sending program to programmable controller).

Per claim 2:

-sending includes sending the one of the first and second processor-executable programs to one of a plurality of intelligent field devices, the intelligent field devices including programmable controllers.

Odem: [0035], sending [0002], industrial control system, safety features [0026], controllers 6c, PLCs 7, devices may exist in any number and in combination with other devices common in industrial control system...actuators, instruments, motors, valves, pumps, fans, etc.

Per claim 3:

-sending includes sending the one of the first and second processor-executable programs to a modular redundant control system, the modular redundant control system including at least two processor modules,

-wherein the modular redundant control system distributes the one of the first and second processor-executable programs to the at least two processor modules,

-and wherein each of the processor modules is configured to carry out instructions encoded in the one of the first and second processor-executable programs.

[[0003], [0026], modular redundant control systems [0017], control language IEC 61131-3 [0026], device and / or the industrial control system comply with safety standards such as Safety Integrity Levels (SIL) as defined in the standard IEC 61508.

Per claim 4:

-the first processor-executable program and the second processor-executable program include

encoded instructions for controlling field devices.

Odem: [0027], A device 6a for safety applications in a process control system typically executes user written applications described in a high level language derived from the standard IEC 61131-3.

Per claim 5:

-the first processor-executable program and the second processor-executable program include operating-system instructions for controlling the programmable controller.

Odem: [0027], A device 6a for safety applications in a process control system typically executes user written applications (operating system instructions for controller device) described in a high level language derived from the standard IEC 61131-3.

Per claim 10:

-the comparing includes comparing the first and second processor-executable programs bit-by-bit.

Odem: [0033], compare CRC, checksum, or parity check [0038], comparison of remainder values

Per claim 11:

-the comparing includes generating a cyclical redundancy code for each of the first and second processor-executable programs, and comparing the cyclical redundancy codes for the first and

second processor-executable programs.

Odem: [0033], CRC

Per claim 13:

-the converting includes converting the source program to the first processor-executable program and asynchronously converting the source program to the second processor executable program.

Odem: [0039], compile test program, compare second software means and first software means, using variable that changes over time.

Per claim 14:

-receiving a reference source program and a reference executable program, wherein the reference executable program includes processor executable code generated on a trusted system from the reference source program;

-converting the reference source program to a test program using system resources used to convert the source program to the one of the first and second processor-executable programs; comparing the reference executable program with the test program;

-validating, in response to the reference executable program and the test program being substantially the same, the system resources used to convert the source program to the one of the first and second processor-executable programs.

Odem: [0029], receive test program (reference source program) generated with certified test equipment (trusted system), validating a compiler for safety control Also see rejection of limitations as addressed in claim 1 above.

Per claim 15:

A system for generating a control program comprising:

-a processor; a network communication module coupled to the processor, wherein the network communication module is configured to communicate with a programmable controller via a network, the programmable controller associated with a safety instrumented function;

-a memory associated with the processor, the memory including:

-conversion code including encoded instructions for separately converting a source program into first and second load modules, the first and second load modules including processor-executable code to control the programmable controller;

-comparison code including encoded instructions for comparing the first and second load modules and sending one of the load modules to the programmable controller in response to the first and second load modules being substantially the same.

See rejection of limitations in claims 1 & 2 above. Also see [0026], industrial control system with a device (programmable memory with processor), a user written program, connected by communication (network communication) means to the device. [0030-0031], conversion code and comparison code.

Per claim 16:

-the programmable controller is an intelligent field device, and wherein at least one other intelligent field devices is coupled to the network.

See rejection of limitations addressed in claim 2 above and [0026].

Per claim 17:

-the programmable controller is a main processor in a modular redundant control system, wherein the sending includes sending the one of the load modules to at least one other programmable controller in the modular redundant control system.

See rejection of limitations addressed in claim 3 above.

Per claim 18:

-the one of the load modules includes encoded instructions for controlling field devices with the programmable controller.

See rejection of limitations addressed in claim 4 above.

Per claim 19:

-the one of the load modules includes operating-system instructions for controlling the programmable controller.

See rejection of limitations addressed in claim 5 above.

Per claim 22:

-the conversion code includes encoded instructions to asynchronously generate the first and second load modules using the processor.

See rejection of limitations addressed in claim 13 above.

Per claim 23:

-including another processor, wherein the conversion code includes encoded instructions to generate the first load module with the processor and the second load module with the other processor.

See rejection of limitations addressed in claim 12 above.

Per claim 27:

-the comparison code includes encoded instructions to compare the first and second load modules on a bit-by-bit basis.

See rejection of limitations addressed in claim 10 above.

Per claim 28:

-the comparison code includes encoded instructions for sending the one of the load modules to the programmable controller in response to the first and second load modules being identically the same.

See rejection of limitations addressed in claims 1, 11, and 14 above.

Per claim 29:

-the comparison module includes instructions to generate cyclical redundancy codes for the first and second load modules, and instructions to compare the cyclical redundancy codes for the first and second load modules.

See rejection of limitations addressed in claim 11 above.

Per claim 30:

- the memory includes: a reference source program; and a reference executable program;
- wherein the conversion code includes encoded instructions to convert the reference source program into a test program using system resources associated with the one of the load modules sent to the programmable controller;
- wherein the comparison code includes encoded instructions for comparing the test program and the reference executable program, and validating, in response to the test program and the reference executable program being substantially the same, the system resources associated with the one of the load modules sent to the programmable controller.

See rejection of limitations addressed in claim 14 above. Also see [0019], memory, [0030-0032], using a test, convert compiler code and compare results of using test on compiler.

Per claim 31:

A processor-readable medium including code to generate an executable program from a source program when carried out by a processor, the executable program being disposed for execution by a programmable controller, the code comprising:

- conversion code for converting a source program into the executable program;
- a code segment for loading two copies of at least a portion of the conversion code into a memory associated with the processor such that the two copies do not occupy the same location in the memory;

-each of the two copies including code to generate a corresponding one of two copies of at least a portion of the executable program;

-comparison code for comparing the two copies of the at least the portion of the executable program, and for sending at least one of the two copies of the at least the portion of the executable program to the programmable controller in response to the two copies of the at least the portion of the executable program being substantially the same.

Odem: [0019], memory / computer program product containing software code [0033-0040], generating compiler code, checking CRC, compiling a second time, comparing the first and second software for errors, and downloading validated code with safety features for control of real world entities.

Per claim 39:

-the comparison code includes code for comparing the two copies of the at least the portion of the executable program on a bit-by-bit basis.

See rejection of limitations addressed in claim 10 above.

Per claim 40:

-the comparison code includes: a code segment for generating a cyclical redundancy code for each of the two copies of the at least the portion of the executable program;

-a code segment for comparing the cyclical redundancy codes for the two copies of the at least the portion of the executable program.

See rejection of limitations addressed in claim 11 above.

Per claim 41:

-a reference source program; a reference executable program, the reference executable program includes processor executable code generated on a trusted system from the reference source program;

-wherein the conversion code includes code for converting the reference source program to a test program using system resources used to convert the at least one of the two copies of the at least the portion of the executable program sent to the programmable controller;

-wherein the comparison code includes code for comparing the reference executable program with the test program and validating, in response to the reference executable program and the test program being substantially the same, system resources used to convert the at least one of the two copies of the at least the portion of the executable program sent to the programmable controller.

See rejection of limitations addressed in claim 14 above.

Per claim 42:

A method for building a program for execution by a programmable controller, the method comprising:

-receiving a source program, the source program including high-level instructions for controlling a programmable controller, the programmable controller being associated with a safety instrumented function;

-converting the source program to a first processor-executable program and a second processor-executable program;

-comparing the first and second processor-executable programs;

-generating a signal indicative of the similarity between the first and second processor-executable programs.

See rejection of limitations addressed in claim 15 above.

Per claim 47:

-the first processor-executable program and the second processor-executable program include encoded instructions for controlling field devices associated with the safety instrumented function.

See rejection of limitations addressed in claim 18 above.

Per claim 48:

-the first processor-executable program and the second processor-executable program include operating-system instructions for controlling the programmable controller.

See rejection of limitations addressed in claim 19 above.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are

such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 6-9, 12, 20, 21, 24-26, 32-38, and 43-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent Application Publication 2006 / 0101433 A1 to Ophem et al., in view of US Patent 5,754,860 to McKeeman et al.

Per claim 6:

- loading a first compiler into a first memory location;
- loading a second compiler into a second memory location separate from the first memory location;
- wherein the converting includes converting the source program to a first piece of intermediate code with the first compiler and converting the source program to a second piece of intermediate code with the second compiler.

Odem: See FIGs. 2 & 3

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors, residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Ophem, to consider testing controller / safety code under as many

different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 7:

-the first and second memory locations are a part of the same computer.

Odem: See FIGs. 2 & 3

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors, residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Odem, to consider testing controller / safety code under as many different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 8:

-the first and second compilers are simultaneously loaded into the first and second memory locations.

Odem: See FIG. 4.

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors, residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Opem, to consider testing controller / safety code under as many different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 9:

Opem failed to explicitly disclose:

-the converting includes storing the first and second pieces of intermediate code in separate disk drives.

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors (separate disk drives), residing in different computer systems, remotely execute and compare results of different

compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Opem, to consider testing controller / safety code under as many different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 12:

Opem failed to explicitly disclose:

-the converting includes converting the source program to the first processor-executable program with a first processor and converting the source program to the second processor-executable program with a second processor.

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors, residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Opem, to consider testing controller / safety code under as many

different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 20:

Opem failed to explicitly disclose:

-the conversion code includes encoded instructions for simultaneously generating the first and second load modules using the processor.

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65).

Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors, residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Opem, to consider testing controller / safety code under as many different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 21:

-random access memory coupled with the processor;

-wherein the conversion code includes encoded instructions to load two copies of a portion of the conversion code into separate portions of the random access memory and simultaneously generate each of the first and second load modules with a corresponding one of the two copies of the portion of the conversion code.

See rejection of limitations addressed in claims 6 and 20 above. Also, [0015], [0019], ensure that no fault is introduced into the industrial control system, error can be due to failure in computer's memory or failure in a disk where the compiler code is stored (random access memory).

Per claim 24:

-a first storage device coupled to the processor; a second storage device coupled to the processor;  
-wherein the conversion code includes encoded instructions to store intermediate code associated with the first load module in the first storage device and store intermediate code associated with the second load module in the second storage device.

Opem: See FIG4, separate storage.

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors (separate storage), residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Opem, to consider testing controller / safety code under as many different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly.

Per claim 25:

Opem failed to explicitly disclose:

-wherein the first and second storage devices are hard drives.

More explicitly, McKeeman disclosed comparing the results using two compilers. A source program processed by multiple compilers are compared for corresponding output (col. 6: 23-65). Col. 6: 58-60, equivalent object files (intermediate) Col. 32: 21-52, second version and second compiler for comparison. Col. 37: 57 – col. 38: 7, networked processors (separate storage hard drives), residing in different computer systems, remotely execute and compare results of different compilers, each residing in a different computer system Col. 39: 5-31, compare runtime output of two versions produced using the same compiler

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Opem, to consider testing controller / safety code under as many different situations as possible, to fully consider possible result variations due to the importance of the code executing correctly. .

Per claim 26:

-the first storage device is a local hard drive and the second storage device is a network hard drive.

See rejection to claim 25 above.

Per claim 32:

-the conversion code includes compiler code for converting the source program into intermediate code, and wherein the code for loading two copies of the at least the portion of the conversion code into the memory includes code for loading two copies of the compiler code into the memory such that the two copies of the compiler code do not occupy the same location in memory.

See rejection of limitations addressed in claim 6 above.

Per claim 33:

-code for loading a first piece of intermediate code generated by one of the two copies of the compiler code into a first memory location and loading a second piece of intermediate code generated by another one of the two copies of the compiler code into a second memory location.

See rejection of limitations addressed in claim 9 above.

Per claim 34:

-each of the first and second memory locations are a part of a corresponding one of a first and

second hard drives.

See rejection of limitations addressed in claims 25 & 26 above.

Per claim 35:

-each of the first and second hard drives are controlled by separate disk controllers.

See rejection of limitations addressed in claim 26 above.

Per claim 36:

-each of the first and second memory locations are a part of a corresponding one of a first and second separate RAM memory locations.

Odem: See FIG. 4.

Per claim 37:

-the first memory location is part of a network hard drive and the second memory location is part of a memory location selected from the group consisting of a RAM memory and a local hard drive.

See rejection of limitations addressed in claim 26 above.

Per claim 43:

-loading a first compiler into a first memory location;  
-and loading a second compiler into a second memory location separate from the first memory location;

-wherein the converting includes converting the source program to a first piece of intermediate code with the first compiler and converting the source program to a second piece of intermediate code with the second compiler.

See rejection of limitations addressed in claim 6 above.

Per claim 44:

-the first and second memory locations are a part of the same computer.

See rejection of limitations addressed in claim 7 above.

Per claim 45:

-the first and second compilers are simultaneously loaded into the first and second memory locations.

See rejection of limitations addressed in claim 8 above.

Per claim 46:

-the converting includes storing the first and second pieces of intermediate code in separate disk drives.

See rejection of limitations addressed in claim 9 above.

Per claim 38:

-each of the two copies of at least a portion of the conversion code are in a corresponding one of two dynamic loaded libraries (DLLs), wherein the code for loading the two copies includes code

for loading the two copies into the memory simultaneously such that the two copies do not occupy the same location in the memory.

See rejection of limitations addressed in claim 8 above.

***Conclusion***

6. Note pertinent prior art:

USPN 6,201,997 B1 Giers disclosed (Abstract) using two processors, using two memory locations (col. 2: 33-55) to redundantly test and compare safety critical control systems, synchronously.

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

05/17/2007

  
Mary Steelman  
Primary Examiner